

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Previously Presented) A memory system, comprising:
a memory divided into a plurality of memory blocks, one or more of the memory blocks configured to receive a plurality of requests; and
a memory access circuit configured to provide one of the plurality of requests to one of the plurality of memory blocks in a clock cycle, wherein each request of the plurality of requests is provided to each of the plurality of memory blocks.
2. (Cancelled).
3. (Original) The memory system of claim 1, wherein the memory access circuit is configured to stall other requests for the memory block during the clock cycle.
4. (Original) The memory system of claim 3, further comprising a local memory, wherein the local memory is configured to store an unsatisfied request.
5. (Original) The memory system of claim 1, wherein the plurality of requests comprises a plurality of new requests.
6. (Original) The memory system of claim 1, wherein the plurality of requests comprises a plurality of previously unsatisfied requests.
7. (Original) The memory system of claim 1, wherein the plurality of requests comprises one or more new requests and one or more previously unsatisfied requests.
8. (Previously Presented) A memory management system, comprising:
a memory divided into a plurality of memory blocks; and

a plurality of memory block management circuits, each of the plurality of memory blocks associated with one of the memory block management circuits, wherein each memory block management circuit is configured to:

receive one or more requests from one or more requesters for one or more of the plurality of memory blocks, and

on a single clock cycle,

determine whether one of the memory blocks is requested by one or more of the requesters,

determine which request is provided to said one of the memory blocks if one of the memory blocks is requested by one or more of the requesters, and

provide the determined request to said one of the memory blocks, wherein the determination of which request is provided to said one of the memory blocks is based on a duration a request is unsatisfied.

9. (Original) The memory management system of claim 8, further comprising a local memory, wherein the memory block management circuits are configured to store in the local memory an unsatisfied request that is not provided to said one of the memory blocks during the single clock cycle.

10. (Original) The memory management system of claim 8, wherein the one or more requests includes a new request.

11. (Original) The memory management system of claim 8, wherein the one or more requests includes a previously unsatisfied request.

12. (Original) The memory management system of claim 8, wherein the one or more requests include a new request and a previously unsatisfied request.

13. (Original) The memory management system of claim 8, the memory comprising a single port memory.

14. (Original) The memory management system of claim 8, the memory comprising a static random access memory.

15. (Original) The memory management system of claim 8, wherein a number of the one or more requests from the one or more requesters is less than a number of the plurality of memory blocks.

16. (Original) The memory management system of claim 8, wherein a number of requests from the one or more requesters is equal to a number of the plurality of memory blocks.

17. (Original) The memory management system of claim 8, wherein a number of requests from the one or more requesters is greater than a number of the plurality of memory blocks.

18. (Original) The memory management system of claim 8, wherein the determination of which request is provided to said one of the memory blocks is based on an arbitration.

19. (Cancelled).

20. (Original) The memory management system of claim 8, wherein the determination of which request is provided to said one of the memory blocks is based on user-defined criteria.

21. (Original) The memory management system of claim 8, wherein the determination of which request is provided to said one of the memory blocks is based on programmable criteria.

22. (Original) The memory management system of claim 8, wherein the determination of which request is provided to said one of the memory blocks is based on a dynamic response to usage patterns.

23. (Original) The memory management system of claim 8, wherein the one or more of the requests includes a priority request.

24. (Original) The memory management system of claim 23, wherein the priority request is determined to access said one of the memory blocks before other requests.

25. (Original) The memory management system of claim 23, the priority request comprising an input/output request.

26. (Original) The memory management system of claim 8, wherein the memory block management circuit further comprises:

an interface configured to receive the one or more requests from the one or more requesters;

an arbiter coupled to the interface, the arbiter configured to:

consider the one or more requests,

determine whether one of the plurality of memory blocks is requested, and

determine which of the one or more requests is provided to said one of the memory blocks; and

a selection circuit configured to route the determined request from the interface to said one of the memory blocks based on the arbiter determinations.

27. (Original) The memory management system of claim 26, the selection circuit comprising a multiplexer.

28. (Original) The memory management system of claim 26, wherein the arbiter determinations are based on request identification data.

29. (Original) The memory management system of claim 26, wherein the determined request data is routed to an input of the selection circuit.

30. (Original) The memory management system of claim 26, the interface further comprising:

a multiplexer with inputs configured to receive

a first request, and

a second request for said one of the memory blocks, wherein the first request precedes the second request, and

a stall signal indicating whether the first request has been satisfied,

the multiplexer configured to select the first or second request based on the stall signal;

and

a latch configured to:

store the unprocessed request, and

output the unsatisfied request to the multiplexer based on the stall signal.

31. (Original) The memory management system of claim 30, wherein the stall signal indicates the first request was not satisfied, wherein the first request is selected before the second request resulting in pipelined data.

32. (Original) The memory management system of claim 30, when the latch is empty and the first request is issued, the multiplexer being configured to select the first request and the latch remains empty.

33. (Original) The memory management system of claim 30, when the latch is empty, wherein the first request is not determined to be satisfied and is stored to the latch.

34. (Original) The memory management system of claim 33, wherein the unsatisfied first request is automatically re-issued from the latch.

35. (Original) The memory management system of claim 34, wherein if the first request continues to be unsatisfied, the unsatisfied first request is retained in the latch.

36. (Original) The memory management circuit of claim 30, wherein the first request is determined to be provided to said one of the memory blocks, and wherein the second request is unsatisfied and reissued by the requester.

37. (Original) The memory management system of claim 30, wherein for each unsatisfied request, a stall signal is provided to:

the requester that issued the unsatisfied request, and

all memory block management block circuits.

38. (Original) The memory management circuit of claim 37, wherein the interface further comprises a delay circuit configured to time the stall signal, the multiplexer being configured to select the first request or the second request based on the stall signal.

39. (Original) The memory management circuit of claim 38, the delay circuit comprising a D flip flop.

40. (Original) The memory management system of claim 8, further comprising a bus configured to transmit the one or more requests to the plurality of memory blocks through respective memory block management circuits.

41. (Original) The memory management system of claim 40, the bus comprises a stall bus, the memory block management circuits being configured to:
receive a stall signal from the stall bus and
prevent the unsatisfied requesters from accessing said one of the memory blocks based on the stall signal.

42. (Original) The memory management system of claim 40, the bus comprising a request bus configured to carry request data from the one or more requesters to the memory management circuits corresponding to a requested memory block.

43. (Original) The memory management system of claim 40, the bus comprising a memory bus configured to carry data retrieved from the requested memory block to a storage element associated with the request.

44. (Original) The memory management system of claim 8, further comprising a plurality of request management circuits, wherein each of the one or more requesters is associated with one of the plurality of request management circuits, and

each of the plurality of request management circuits are configured to interface between a memory bus carrying data retrieved from said one of the memory blocks and the one or more requesters.

45. (Original) The memory management system of claim 44, each of the request management circuits comprising:

a control circuit configured to receive:

a stall signal generated by the memory management circuit of the requested memory block, and

an address signal, read from the one or more requests, identifying the requested memory block,

the control circuit being configured to generate a selection signal based on the stall signal and the address signal; and

a multiplexer configured to receive the selection signal and select bus lines carrying data requested by the one or more requesters.

46. (Original) The memory management system of claim 45, the address signal comprising a binary number identifying a requested memory block.

47. (Original) The memory management system of claim 45, the control circuit further comprising a delay system configured to synchronize the selection signal to pass the requested data through the multiplexer.

48. (Previously Presented) The memory management system of claim 47, the delay system comprising:

a first delay circuit configured to receive the stall signal;

a second delay circuit configured to receive the address signal;

a third delay circuit configured to provide the address signal to said multiplexer; and

a delay multiplexer configured to select one of two or more inputs based on an output of the first delay circuit, the inputs to the delay multiplexer comprising:

an output of the second delay circuit, and

an output of the third delay circuit.

49. (Original) The memory management system of claim 48, the first delay circuit comprising a D flip flop.

50. (Original) The memory management system of claim 48, the second delay circuit comprising a series of D flip flops, wherein

a number of D flip flops represents a binary number, and

each of the plurality of memory blocks is associated with the binary number.

51. (Original) The memory management system of claim 48 wherein a common clock is configured to drive the first, second, and third delay circuits.

52. (Original) The memory management system of claim 48, wherein the stall signal and the address signal are passed through the first and second delay circuits on the same clock.

53. (Original) The memory management system of claim 8, wherein a request to write data to one of the plurality of memory blocks is completed in at least two clocks.

54. (Original) The memory management system of claim 8, wherein a request to read data from one of the plurality of memory blocks is completed in at least three clocks.

55. (Currently Amended) A method of processing requests for memory, comprising:
during a first clock,
driving a first set of one or more requests from one or more requesters onto a request bus,
performing a first determination whether one of a plurality of memory blocks is requested by the one or more requests, and
if two or more requesters issue requests for one of the plurality of memory blocks,
performing a second determination of which request of the two or more requests is provided to said one of the memory blocks, wherein the second determination being based on user-defined criteria.

56. (Original) The method of claim 55, further comprising matching the one or more requests to the plurality of memory blocks by interleaving.

57. (Original) The method of claim 55, if one requester requests one of the plurality of memory blocks, further comprising providing the request to the requested memory block during the first clock.

58. (Original) The method of claim 55, if two or more requests request one of the plurality of memory blocks, further comprising providing one of the two or more requests to the requested memory block based on the first and second determinations.

59. (Previously Presented) The method of claim 55, wherein driving the first set of one or more requests further comprises driving a new request onto the request bus.

60. (Previously Presented) The method of claim 55, wherein driving the first set of one or more requests further comprises driving a previously unsatisfied request onto the request bus.

61. (Previously Presented) The method of claim 55, wherein driving the first set of one or more requests further comprises driving a new request and a previously unsatisfied request onto the request bus.

62. (Original) The method of claim 55, during a second clock, if a request of the one or more requests is to write data to said one of the memory blocks, further comprising latching data of the determined request to said one of the memory blocks.

63. (Original) The method of claim 55, the second determination being based on an arbitration.

64. (Original) The method of claim 55, the second determination being based on prioritizing one request relative to other requests.

65. (Original) The method of claim 64, further comprising granting priority to an input/output request.

66. (Original) The method of claim 64, further comprising granting priority to a previously unsatisfied request.

67. (Cancelled).

Appln No. 10/002,449
Amdt date July 29, 2005

68. (Original) The method of claim 55, the second determination being based on programmable criteria.

69. (Original) The method of claim 55, the second determination being based on a dynamic response to usage patterns.

70. (Original) The method of claim 55, further comprising:
during a second clock,

issuing a stall signal to:

the requester that issued an unsatisfied request, and
the memory management circuits of respective memory blocks.

71. (Original) The method of claim 70, wherein a duration of the stall signal comprises a number of clocks less than or equal to a number of requesters.

72. (Original) The method of claim 70, the second determination being based on prioritizing one request relative to other requests, wherein a maximum duration of the stall signal is based on criteria other than the number of requesters.

73. (Original) The method of claim 70, further comprising storing the unsatisfied request to a local memory.

74. (Original) The method of claim 70, during the second clock, further comprising:
invoking for consideration any requests that were not satisfied from respective local memories, and
invoking a second set of one or more new requests from one or more requesters.

75. (Original) The method of claim 74, further comprising:
performing a first determination whether one of the plurality of memory blocks is requested by one or more requests that were not satisfied and one or more new requests; and
performing a second determination of which request of the one or more requests is provided to said one of the memory blocks.

76. (Original) The method of claim 75, wherein the one or more requests includes a new request.

77. (Original) The method of claim 75, wherein the one or more requests includes an unsatisfied request.

78. (Original) The method of claim 55, during a second clock, if the request is a request to read data from the requested memory block, further comprising:

providing data retrieved from said one of the memory blocks onto a memory bus directed to the determined requester,

wherein the memory block from which the retrieved data is stored is identified by a destination address retrieved from the determined requester while driving the first set of one or more requests onto the request bus during the first clock.

79. (Previously Presented) The method of claim 78, during a third clock, further comprising latching the requested data retrieved from said one of the memory blocks internally to the determined requester.

80. (Original) The method of claim 55, wherein the one or more requests are issued from a direct memory access device.

81. (Original) The method of claim 55, wherein the one or more requests are issued from a processor.

82. (Original) The method of claim 55, wherein the one or more requests are issued from an input/output device.

83. (Original) A method of writing pipelined data to a single port memory, comprising: during a first clock,

receiving a set of requests for one or more of the memory blocks,
determining whether one of the memory blocks is requested by one or more requests, and
determining which request is provided to said one of the memory blocks; and

during a second clock,
latching data of the determined request to said one of the memory blocks,
accessing said one of the memory blocks, and
writing data of the determined request to said one of the memory blocks.

84. (Original) A method of reading pipelined data from a single port memory, comprising

during a first clock,
receiving a plurality of requests for one or more of the memory blocks,
determining whether one of the memory blocks is requested by one or more requests of the plurality of requests, and
if one or more of the memory blocks is requested by one or more requests, determining which request is provided to said one of the memory blocks;
during a second clock,
latching data of the determined request to said one of the memory blocks,
accessing said one of the memory blocks, and
retrieving requested data from said one of the memory blocks; and
during a third clock,
driving the retrieved data from said one of the memory blocks requested data onto a return bus,
routing the retrieved data to the determined requester, and
during a fourth clock,
latching the retrieved data to the determined requester.

85. (Original) A memory management system comprising:
during a first clock,
means for determining whether one of the plurality of memory blocks is requested by one or more requests of a first plurality of requests, and
means for determining which request of the one or more requests of the first plurality of requests is provided to said one of the memory blocks.

86. (Original) A system for writing pipelined data to a single port memory, comprising:
during a first clock,

Appln No. 10/002,449
Amdt date July 29, 2005

means for receiving a set of requests for one or more of the memory blocks,
means for determining whether one of the memory blocks is requested by one or more requests, and
if one or more of the memory blocks is requested by one or more requests, means for determining which request is provided to said one of the memory blocks; and
during a second clock,
means for latching data of the determined request to said one of the memory blocks,
means for accessing said one of the memory blocks, and
means for writing data of the determined request to said one of the memory blocks.

87. (Original) A system for reading pipelined data from a single port memory, comprising:

during a first clock,
means for receiving a plurality of requests for one or more of the memory blocks,
means for determining whether one of the memory blocks is requested by one or more requests of the plurality of requests, and
if one or more of the memory blocks is requested by one or more requests, means for determining which request is provided to said one of the memory blocks;
during a second clock,
means for latching data of the determined request to said one of the memory blocks,
means for accessing said one of the memory blocks, and
means for retrieving requested data from said one of the memory blocks; and
during a third clock,
means for driving the retrieved data from said one of the memory blocks requested data onto a return bus,
means for routing the retrieved data to a memory corresponding to the determined requester, and
during a fourth clock,
means for latching the retrieved data to the determined requester.